Тестирование безопасности

План лекции

1. Введение.
2. Принципы безопасности ПО.
3. Что проверять?
4. Виды уязвимостей.
5. Как тестировать ПО на безопасность.
6. Инструменты.

Тестирование безопасности - это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.

Принципы безопасности ПО

- **Конфиденциальность** - это сокрытие определенных ресурсов или информации.
- **Целостность** – состоит из двух критериев: Доверие и Повреждение и восстановление.
- **Доступность** - требования о том, что ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству.

Что проверять?

- Контроль доступа
- Аутентификация
- Валидация входных значений
- Криптография
- Механизмы обработки ошибок
- Конфигурация сервера
- Интеграция со сторонними сервисами
- Проверка устойчивости к Dos/DDos атакам

OWASP

Open Web Application Security Project (OWASP) — это открытый проект обеспечения безопасности веб-приложений.

https://www.owasp.org/

Виды уязвимостей

- **XSS (Cross-Site Scripting)** - это вид уязвимости программного обеспечения (Web приложений), при которой, на генерированной сервером странице, выполняются вредоносные скрипты, с целью атаки клиента.
- **XSRF / CSRF (Request Forgery)** - это вид уязвимости, позволяющий использовать недостатки HTTP протокола.
- **Code injections (SQL, PHP, ASP и т.д.)** - это вид уязвимости, при котором становится возможно осуществить запуск исполняемого кода с целью получения доступа к системным ресурсам, несанкционированного доступа к данным либо выведения системы из строя.
- **Server-Side Includes (SSI) Injection** - это вид уязвимости, использующий вставку серверных команд в HTML код или запуск их напрямую с сервера.
- **Authorization Bypass** - это вид уязвимости, при котором возможно получить несанкционированный доступ к учетной записи или документам другого пользователя.

Как тестировать на безопасность?

1. Google? А почему бы и нет?
2. Для проверки на XSS разместить на странице скрипт, например: <script>alert(document.cookie);</script>
3. Наиболее частыми CSRF атаками являются атаки использующие HTML <IMG> тэг или Javascript объект image. Наример: <img src="http://hacker_site/?command">

4. Code injections

SQL-запрос на сервер: SELECT Username
FROM Users
WHERE Name = 'tester'
AND Password = 'testpass';

Вводимые данные: имя ’tester’
пароль testpass' OR '1'='1

Итоговый запрос: SELECT Username
FROM Users
WHERE Name = 'tester'
AND Password = 'testpass' OR '1'='1';

5. Команда, которая выводит на экран список файлов в OS Linux: < !--#exec cmd="ls" -->

6. Для проверки на уязвимость Authorization Bypass попробуйте подставить вместо своего userID в адресе страницы личного профиля номер другого пользователя.

Инструменты

Сканеры безопасности:

- XSpider, Zenmap, Metasploit – сетевые сканеры, для тестирования уязвимостей, присущих сетевой инфраструктуре.
- Acunetix Web Vulnerability Scanner, XSpider, MaxPatrol, инструментарий OWASP Live CD – специализированный набор инструментов для тестирования безопасности и логики работы web-приложения.

Ручное и полуавтоматизированное тестирование безопасности:

- Intercepter-NG, WinDump, WireShark и др. – снифферы для перехвата и анализа сетевого траффика.
- FireBug, Web Developer – плагины для Firefox, которые можно использовать для изменения логики работы клиентской части приложения.

- **Tamper Data** – простой, быстрый и эффективный инструмент, который используется при проведении испытания на возможность проникновения в систему.
- **SkipFish** - бесплатный сканер безопасности с открытым кодом.
- **Wapiti** выполняет сканирование методом «чёрного ящика» и вводит полезные данные, чтобы проверить, уязвим ли сценарий.
- **SQLMap** - бесплатный сканер с открытым исходным кодом, главная задача которого автоматизированный поиск SQL уязвимостей.
- **RIPS** — сканер предназначен для отслеживания «узких» мест, статичного кода PHP.